Version Control Software

Submitted by:

Dileep Kumar Maurya(15IT208)

A component of software configuration management, **version control**, also known as **revision control** or **source control**, is the management of changes to documents, computer programs, large web sites, and other collections of information. Changes are usually identified by a number or letter code, termed the "revision number", "revision level", or simply "revision". For example, an initial set of files is "revision 1". When the first change is made, the resulting set is "revision 2", and so on. Each revision is associated with a time-stamp and the person making the change. Revisions can be compared, restored, and with some types of files, merged.

The need for a logical way to organize and control revisions has existed for almost as long as writing has existed, but revision control became much more important, and complicated when the era of computing began. The numbering of book editions and of specification revisions are examples that date back to the print-only era. Today, the most capable (as well as complex) revision control systems are those used in software development, where a team of people may change the same files.

Top 3 Version control Softwares:

1. Mercurial

This is yet another form of version control system, similar to Git. It was designed initially as a source for larger development programs, often outside of the scope of most system admins, independent web developers and designers. However, this doesn't mean that smaller teams and individuals can't use it. Mercurial is a very fast and efficient application. The creators designed the software with performance as the core feature.

Aside from being very scalable, and incredibly fast, Mercurial is a far simpler system to use than things such as Git, which one of the reasons why certain system admins and developers use it. There aren't quite many things to learn, and the functions are less complicated, and more comparable to other CVS systems. Mercurial also comes alongside a web-interface and various extensive documentation that can help you to understand it better.

Mercurial offers following distinctive features:

- Distributed Architecture
- Fast
- Platform Independent
- Extensible
- Easy to use
- Open Source

2. SVN

SVN, or Subversion as it is sometimes called, is generally the version control system that has the widest adoption. Most forms of open-source projects will use Subversion because many other large products such as Ruby, Python Apache, and more use it too. Google Code even uses SVN as a way of exclusively distributing code.
Because it is so popular, many different clients for Subversion are available. If you use Windows, then Tortoise SVN may be a great browser for editing, viewing and modifying Subversion code bases. If you're using a MAC, however, then Versions could be your ideal client.

3. GIT

Git is considered to be a newer, and faster emerging star when it comes to version control systems. First developed by the creator of Linux kernel, Linus Torvalds, Git has begun to take the community for web development and system administration by storm, offering a largely different form of control. Here, there is no singular centralized code base that the code can be pulled from, and different branches are responsible for hosting different areas of the code. Other version control systems, such as CVS and SVN, use a centralized control, so that only one master copy of software is used.
As a fast and efficient system, many system administrators and open-source projects use Git to power their repositories. However, it is worth noting that Git is not as easy to learn as SVN or CVS is, which means that beginners may need to steer clear if they're not willing to invest time to learn the tool.

Git offers the following features:

- Branching and Merging
- Small and Fast
- Distributed
- Data Assurance
- Staging Area
- Free and Open Source
- Trademark

I will be using Git as the version control software.